

第5章

PCI Expressを組み込み分野に広げる拡張規格ASI

——トポロジに柔軟性を持たせるプロトコルの基礎とシステム構成

横川与英

ここでは，PCI Expressの拡張規格の一つである「ASI（Advanced Switching Interconnect）」を取り上げる．ASIは，PCI Expressの物理層を利用し，組み込み機器への展開を見込んでトポロジ構成やプロトコルなどを改良している．例えば，スイッチ・ファブリック技術を用いることで，複数のPCI Expressデバイスを同時に通信させたり，冗長性を持たせた接続が可能となる．（編集部）

最近，市販のパソコンにPCI Expressスロットが搭載されるようになりました．今後PCI Expressは，組み込み機器にも採用されていくものと考えられます．従来のバスと比べて広帯域で，少ないピン数による接続が可能なPCI Expressは，多チャネルのI/Oを接続したり，画像処理装置のように大量のデータを分散処理する場合のインターフェースとして適しています．

しかし，PCI Expressのアーキテクチャは，PCIのシングル・アドレス空間を構成する（ルート・コンプレックスは一つのみ）など，マルチプロセッシングを必要とする処理システムを構成する場合には，解決すべきいくつかの問題があります．

そこで生まれたのがASI（Advanced Switching Interconnect）です．ASIはPCI Expressとの互換性を保ちながら，マルチプロセッシングを必要とする高機能・高性能な組み込み機器の構成を可能にします．

1 ASIによるシステム開発のメリット

ASIは，特定用途向けに開発されたインターフェース規格ではありません．その応用範囲は，産業機器から通信インフラ（サーバ，ストレージ，ルータ，メディア・ゲートウェイなど）まで，幅広い分野を想定しています．

ここでは，ASIを利用してPCI Express対応装置の処理能力を向上させる例を紹介します．

対象とする装置の構成を**図1(a)**に示します．この装置の画像入力チャネルを増やすためには，例えば**図1(b)**のようにCPU（ルート・コンプレックス）にx4リンク4組分のPCI Expressポートが必要となります．かりにポートを用意できたとしても，CPUは従来の処理能力の2倍を必要とするため，分散処理を行わなければなりません．

図2にASIを用いた分散処理構成の例を示します．ここでは，PCI Expressの構成を保ったまま移行するため，後

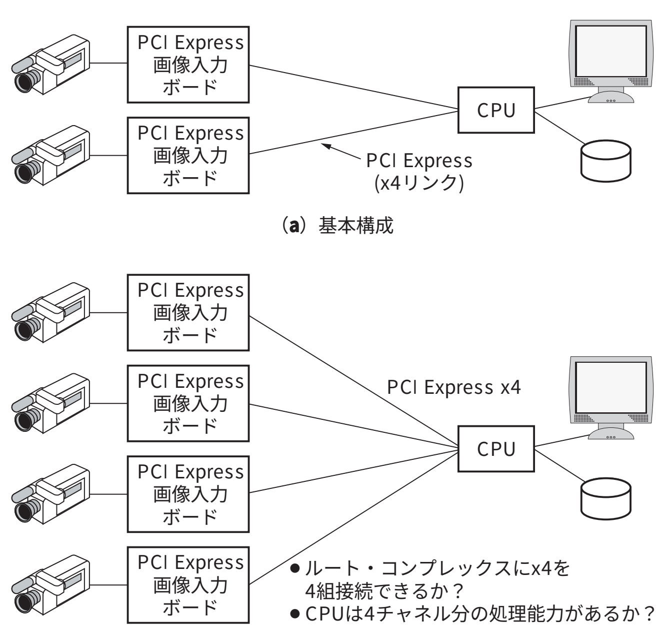


図1　PCI Expressによるシステム構成

対象とする装置は，(a)のようにPCI Expressのx4リンクを持つ画像入力ボード二つをCPUボードと接続し，CPUボードで最大限の画像処理を行うものとする．

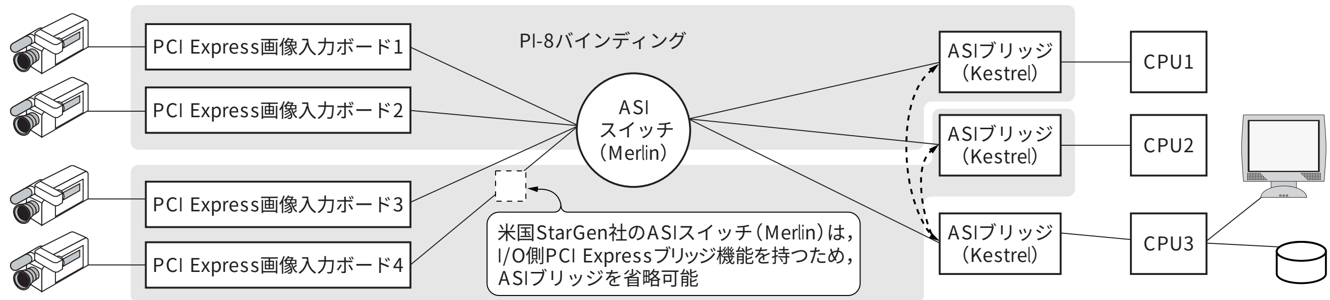


図2　ASIによるシステム構成例
PCI Expressの構成を維持しつつ移行するため，PI-8バインディングを用いてCPUと画像入力ボード2組分をファブリック構成にした．

述のPI-8バインディングというASI特有のしくみを用いてCPUと画像入力ボード2組分をファブリック構成にしています．このようにすることで，画像入力と画像処理にかかわるドライバ，およびアプリケーション・ソフトウェアについては，従来のものをそのまま流用できます．

一方，分散処理としたため，2組のCPUの処理結果をまとめて表示し，記録する必要があります．**図2**では第3のCPUを用意し，それぞれのCPUの間をASI独自のプロトコルを用いてデータ転送しています．また，IPoASI(IP over ASI)を用いることで，アプリケーションはASIを意識することなく，TCP/IPをASI上で使用できます．

さらに，画像入力ボードとCPUの数が増えても，アーキテクチャを変えることなくシステムを拡張できます．

● ASI対応製品は2006年から出荷開始の予定

現在ASI-SIG(ASI Special Interest Group)メンバ企業により，各種ASI対応デバイス，IPコア，ボード製品，プロトコル・アナライザなどの開発が進められています．2006年にはASI製品が量産出荷される見込みです．

2005年6月のSuperCommと同年8月のIntel Developer Forumでは米国Agilent Technologies社，米国StarGen社など，複数のメンバ企業のブースで開発製品の展示や実機によるデモンストレーションが行われました．このように，各社においてASIスイッチや各種プロトコル対応ブリッジなどの開発が進んでおり，今後PCI Expressとともに普及が期待されています(**写真1**)．

2 PCI Expressで実現できないこと

ASIは，多数のプロセッサの間，またはプロセッサとI/Oの間を接続したり，QoS(quality of service)や動的な再構成が実現可能で多様なプロトコルを収容できる接続技術として，ASI-SIGが開発しました．ASIはPCI Express技術を生かしつつ，その機能を強化することによって，現存するPCI Expressシステムを十分に活用できるように考慮されています．

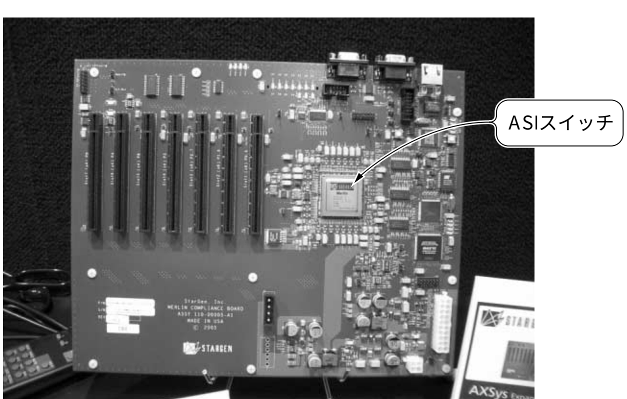


写真1　StarGen社の評価用拡張ボックスのメイン・ボード
ボード上に，ASIスイッチと7本のPCI Expressソケットが見える．

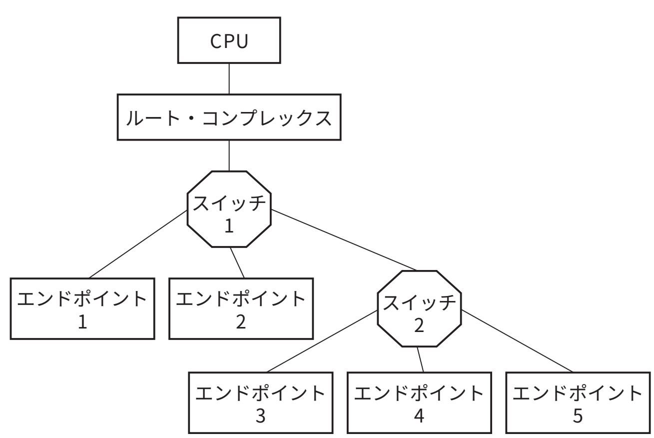


図3　PCI Expressのツリー構造
PCIアーキテクチャをとっているため，ルート・コンプレックスからのツリー構造となる．

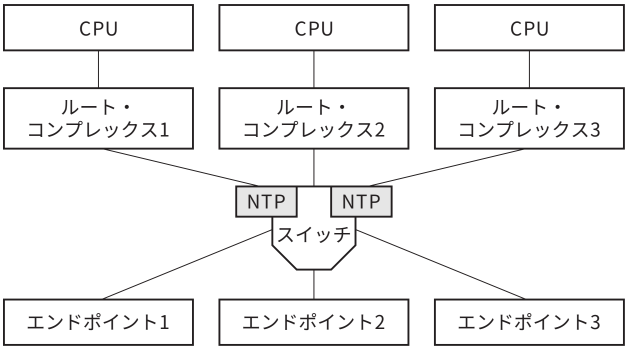


図4 NTPによるマルチプロセッサ接続
PCI Expressでマルチプロセッサ環境を構築する場合，NTPを持つスイッチを使用する．

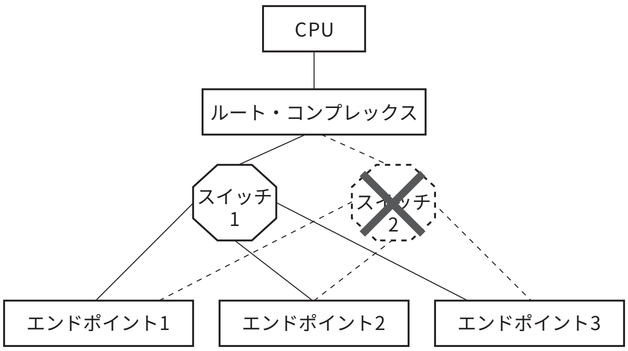


図5 PCI Expressでは実現できない冗長接続
PCI Expressでは，ルート・コンプレックスとエンドポイントの間の接続を二重化することは考慮されていない．

PCIやPCI Expressに関する仕様をPCI-SIGが策定し，管理しているのと同様に，ASIの仕様は米国Intel社やStarGen社をはじめとする多数の半導体メーカや機器メーカからなるASI-SIGにて策定や管理が行われています．ASI-SIGでは，単にASI対応デバイスどうしの通信にかかわる仕様だけでなく，ASIファブリックのための管理ソフトウェアについても標準化が進められています（ASI-SIGのWebサイトはhttp://www.asi-sig.org/）．

●「ツリー構造で単一アドレス空間」がネックに

PCI Expressは，PCIのバスをシリアル化し，ブリッジをスイッチ化することで，デバイス間のデータを高速に転送できるようにしました．しかし，PCIアーキテクチャであるため，その接続形態は**図3**のようにルート・コンプレックスからのツリー構造をとること，そしてそのアドレス空間は単一であることに変わりはありません．

PCI Expressでマルチプロセッサ環境を構築する場合，**図4**のようにNTP（Non-Transparent Port）を持つスイッチを使用して，複数のPCIアドレス空間をPCI Expressに接続する方法が考えられます．その場合，ルート・コンプレックスとNTPの先にある第2，第3のルート・コンプレックス以外のエンドポイントのすべては単一のアドレス空間に配置され，NTPを境に分割されるアドレス空間とは分離されます．また，NTPについてはPCI Express仕様の中でレジスタ構成などが明確に定義されていないため，ソフトウェア互換性の問題が発生することも考えられます．

そのほか，システムの可用性（耐故障性）を高めるために**図5**のようなルート・コンプレックスとエンドポイントの間の接続を二重化することも考慮されていません．

5

3 PCI ExpressとASIの違い

ここでは，PCI ExpressとASIの違いやパケットのルーティング手法など，ASIの基本的な特徴を解説します．

● スター型やメッシュ型のトポロジに対応できる

ASIファブリックは，PCI Expressと同じように，スイッチとエンドポイント，およびそれらを接続するリンクから

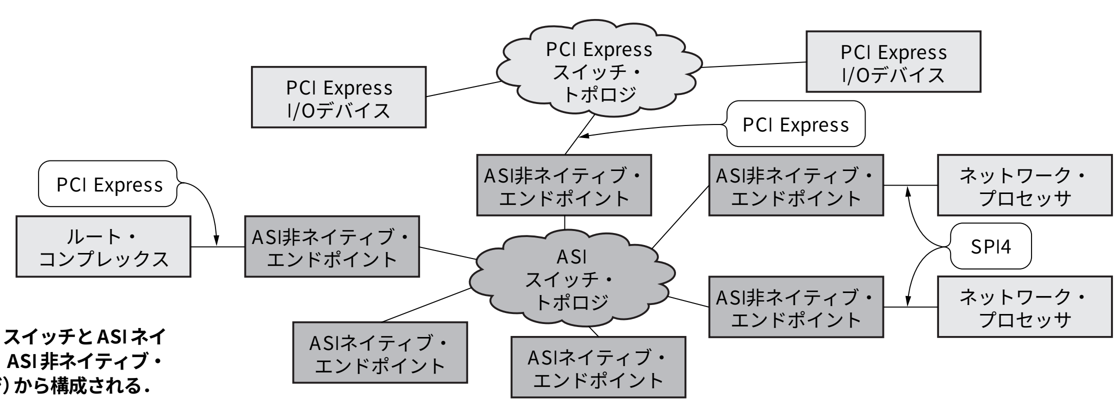
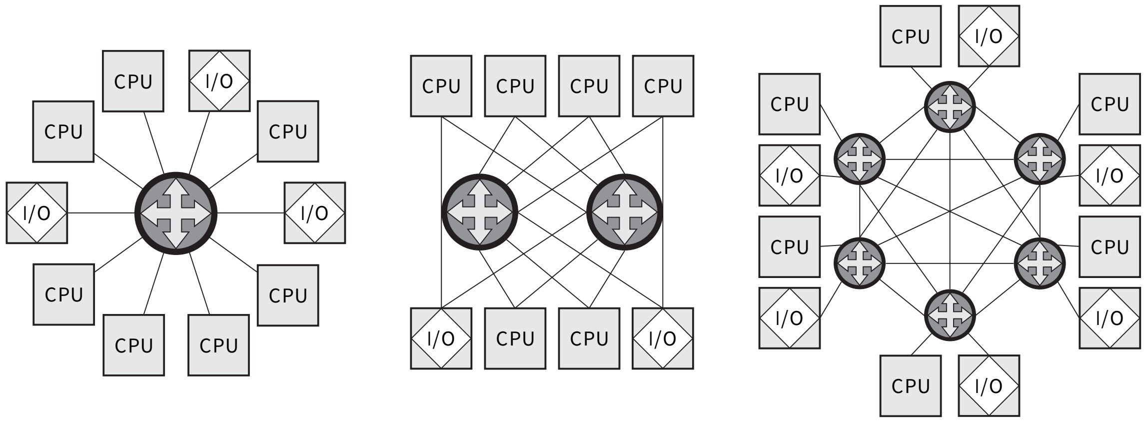


図6
ASIファブリック構成
ASIファブリックは，ASIスイッチとASIネイティブ・エンドポイント，ASI非ネイティブ・エンドポイント（ブリッジ）から構成される．



（**a**）スター　（**b**）デュアルスター　（**c**）フルメッシュ

図7
ASIトポロジ
ASIファブリック内のスイッチは，エンドポイントがPCI Expressへのブリッジであってもツリー構造を持つ必要はない．

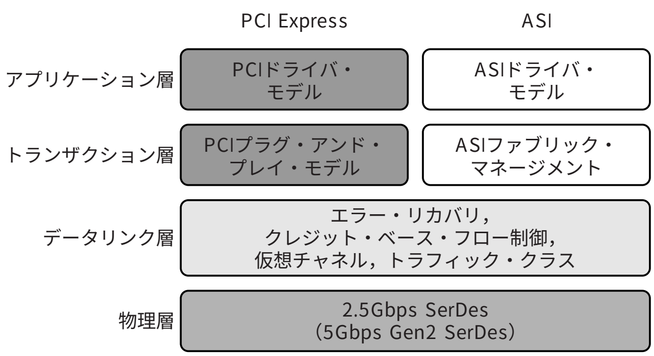
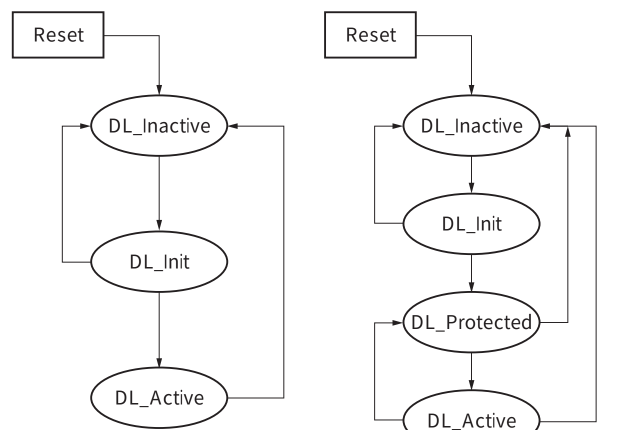


図8　ASI階層モデル
ASIは，物理層とデータリンク層にPCI Express技術を用いる．



（**a**）PCI Expressのリンク・ステート・マシン　（**b**）ASIのリンク・ステート・マシン

図9　リンク・ステート・マシーン
ASIは，PCI Expressのリンク・ステート・マシンにマルチプロセッシング環境を考慮したリンク・ステートを追加している．

構成されます．**図6**に示すように，エンドポイントにはASIネイティブ・エンドポイントと，ASI非ネイティブ・エンドポイントがあります．ASIネイティブ・エンドポイントでは，そこでASIパケットの処理を行います．一方，ASI非ネイティブ・エンドポイントは，PCI ExpressやSPI4など，ASI以外のプロトコルへのブリッジの役目を果たします．

ASIファブリック内のスイッチは，エンドポイントがPCI Expressへのブリッジであっても，ツリー構造を持つ必要はなく，**図7**のように自由なトポロジを組むことができます．

● PCI Expressの物理層とデータリンク層を利用する

図8に，ASIの階層モデルをPCI Expressと対比して示します．ASIのトランザクションは，PCI Expressの物理層とデータリンク層を用いて行います．PCI Expressの物理層とリンク層を採用した理由として，今後の普及が期待できること，普及とともに安定したSerDes（serializer deserializer）によって相互接続性を確保できること，対応チップやSerDesのIP（intellectual property）コアのコストダウンが見込めることなどが挙げられます．

また，ASIは，トランザクション層を含めてPCI Expressとの互換性を保っています．このため，ASIを採用する際に，既存のPCI Expressのリソース（ハードウェア，ソフトウェア，ノウハウなど）を流用して，PCI ExpressからASIへの移行を容易にすることも考慮されています．

● リンク・ステート・マシンと仮想チャネルを改良

ASIはPCI Expressのデータリンク層を用いていると述べましたが，厳密にはマルチプロセッシング環境を考慮した改良が施されています．

PCI Expressリンクの両端のデバイスは，リンクに必要な初期化（DL_Init）が終了すると，自動的にアクティブ状態

表1　ASIの仮想チャネル

UVC(unicast VC)	BVC(bypass capable VC)	1～8本
	OVC(ordered only VC)	0～8本
MVC(multicast VC)		0～4本

ビット	フィールド
31～25	ヘッダCRC
24～20	ターン・ポインタ
19	FECN
18～14	クレジット・リクエスト
13	TS
12	OO
11～9	トラフィック・クラス
8	PCRC
7	P
6～0	プロトコル・インターフェース
31 (2ワード目)	D
30～0 (2ワード目)	ターン・プール

D：direction　FECN：forward explicit congestion notification
OO：ordered-only　P：periahable　PCRC：packet CRC　TS：type specific

図11　ASIルート・ヘッダ

ASIルート・ヘッダには，パケットのルーティング情報のほか，パケットが扱うプロトコル種別などの属性情報が含まれる．

(DL_Active)に遷移して，デバイス間でトランザクションを流せるようになります．マルチプロセッシング環境では，自動的にDL_Activeになることが問題となる場合があります．例えば，プロセッサ間がPCI Expressで接続されていたとします．このとき，機器のリンクが成立してDL_Activeになると，ある任意のCPUから見ると，相手のプロセッサが自由にローカルのアドレス空間内にアクセスできることになります．これは，セキュリティ上の問題があります．

ASIでは，PCI ExpressのDL_InitとDL_Activeの間にDL_Protectedという状態を追加しました．DL_Protectedでは，ASIファブリックを管理するための特別なトランザクションのみを流すことができ，通常のデータ・アクセスは禁止状態になります．そして，通常はファブリック管理を行うソフトウェアによって制御します(**図9**)．

次に改良された点は，仮想チャネル(VC：virtual channel)の種類と数の追加です．**表1**に示すように，ASIではPCI Expressと同じ仮想チャネル(ASIでは「BVC：bypass capable virtual channel」と呼ぶ)を最大8本，OVC(ordered-only virtual channel)を最大8本，MVC(multicast virtual channel)を最大4本追加しました．

OVCは，おもにメッセージングのように転送順を保証しなければならないトランザクションに用います．MVCは，スイッチ内のあるポートから受信したパケットを複数の異なるポートへ転送するマルチキャストを実現するための仮想チャネルです．パケットの転送元と転送先が1対多であるMVCに対して，1対1であるBVCとOVCはUVC

ビット	フィールド
31～24	
23～16	Start TLP Kコード(K27.7)
15～12	―
11～0	TLPシーケンス番号
31～0	ASIルート・ヘッダ(dword 0)
31～0	ASIルート・ヘッダ(dword 1)
31～0	PIヘッダ(dword 0)
	・・・
31～0	PIヘッダ(dword *N*)[オプション]
31～0	PIパケット・ペイロード(dword 0)[オプション]
	・・・
31～0	PIパケット・ペイロード(dword *N*)[オプション]
31～0	ASIパケットCRC(PCRC)[オプション]
31～0	データリンク層CRC(LCRC)
31～24	End TLP Kコード(K29.7)

図10　ASIパケット構造

ASIパケットは，PCI Expressの物理層とデータリンク層が付加するスタート，シーケンス番号，LCRC(link CRC)，ASIの物理層が生成するASIルート・ヘッダ，PI(protocol interface)ヘッダ，PIパケット・ペイロード，ASIパケットCRCにより構成される．

(unicast virtual channel)とも呼ばれます．

● 送信元からの相対位置でパケットの伝達先を決める

ASIでは，パケットのルーティング手法として，ソース・ベース・ルーティングを行います．この手法は，送付元(ソース)からの相対位置情報を用いて送付先までパケットの伝達を行うものです．PCI Expressのように，送付先の絶対位置情報(バス番号，デバイス番号，アドレス)を用いません．そのため，任意の一つの送付先に対して複数のパケット伝達経路(パス)を用意すれば，冗長ファブリックを構成できます．また，ファブリック内のスイッチも，最終送付先であるエンドポイントを意識する必要がないため，高速にスイッチングできます．

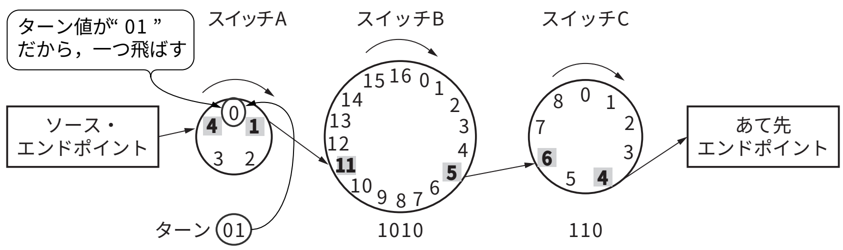
送信元に対して単一の送付先にパケットを伝達することを「ユニキャスト・パス・ルーティング」，送信元に対して複数の送付先にパケットを伝達することを「マルチキャスト・パス・ルーティング」と呼びます(**図10**，**図11**)．

1) ユニキャスト・パス・ルーティング

ユニキャスト・パス・ルーティングのしくみとASIルート・ヘッダのルーティングの一例を**図12**に示します．

ファブリック内の各スイッチは，受信したASIパケットのルート・ヘッダにあるターン・プールとターン・ポインタを参照します．ターン・ポインタの位置からLSB(least

5



転送順	ターン・ポインタ値 入力	ターン・ポインタ値 出力	ターン・プール
ソース・エンドポイント出力	—	9	00000000000000000000000011010110
			9ビット目から2ビット
スイッチA参照値と出力	9	7	00000000000000000000000**01**1010110
スイッチB参照値と出力	7	3	0000000000000000000000001**1010**110
スイッチC参照値と出力	3	0	00000000000000000000000011010**110**
あて先エンドポイント入力	0	—	00000000000000000000000011010110

図12　ユニキャスト・パス・ルーティング（フォワード）の例

ターン値は，パケットの出力ポートを入力ポートからの相対値として示している．スイッチは，パケットを出力する際，ターン・ポインタ値から参照したビット数分を引き去り，次のスイッチが参照するべきビット位置を書き込む．

31〜25	24〜20	19	18〜14	13〜12	11〜9	8	7	6〜0
ヘッダCRC	ターン・ポインタ	FECN	クレジット・リクエスト	00b	トラフィック・クラス	PCRC	P	PI (0000000b)
R	ターン・プール							
マルチキャスト・グループ・インデックス				元データ			セカンダリPI	

図13　マルチキャスト・ルート・ヘッダ

マルチキャスト・パス・ルーティングでは，ルート・ヘッダに続くPIヘッダ内にあるマルチキャスト・グループ・インデックスを参照する．

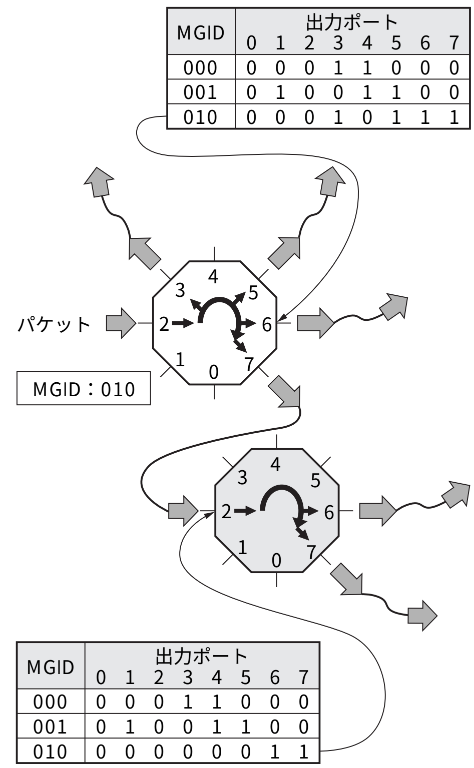


図14　マルチキャストの一例

マルチキャスト・グループID（MGID）を参照するスイッチ内のルックアップ・テーブルから出力するポートを参照し，パケットを複製して送信する．

significant bit）に向かってビット・フィールドを参照し，出力先のポートを認識します．このときビット・フィールドの値はポート番号をインクリメントする方向に何ポート分スキップするのかを示しています．そして，パケットを出力する際には，参照したビット数の分だけターン・ポインタの値を引き去ります．これを繰り返してエンドポイントまでパケットが到達します．もしも，エンドポイントが受け取ったASIパケットのターン・ポインタの値が‘0’でなければ，ルーティング・エラーと判断されます．

ASIパケットがファブリック内を伝播する途中で，なんらかの障害によりあるスイッチから出力できない場合，ASIルート・ヘッダのディレクション（D）ビットをバックワードにし，フォワード・ルーティングと逆方向にターン・プールを参照します．こうすることで，ASIパケットは送り出してきたソース・エンドポイントにパケットを戻すことができます．このとき，パケットに障害情報などを付加することで，ソース・エンドポイントはパケット伝送中にエラーが発生したことを認知できます．

2）マルチキャスト・パス・ルーティング

マルチキャスト・パス・ルーティングの場合，ASIルート・ヘッダのルーティング情報は用いません．ASIルート・ヘッダのプロトコル・インターフェース（PI）が‘0’の場合，ルート・ヘッダに続くPIヘッダ内にあるマルチキャスト・グループ・インデックスを参照して，ルックアップ・テーブルから受信したパケットを送出するポートを判断します（**図13**，**図14**）．

マルチキャスト・パス・ルーティングによって伝達されるパケットの実質的なプロトコルは，マルチキャスト・ルート・ヘッダのセカンダリPIフィールドで示されます．

ルックアップ・テーブルは，最大で65,536種のマルチキャスト・パターンを保持します（実際のデバイスに実装される数はこれより少ない場合がある）．ルックアップ・テーブルは，あらかじめ設定しておく必要がありますが，通

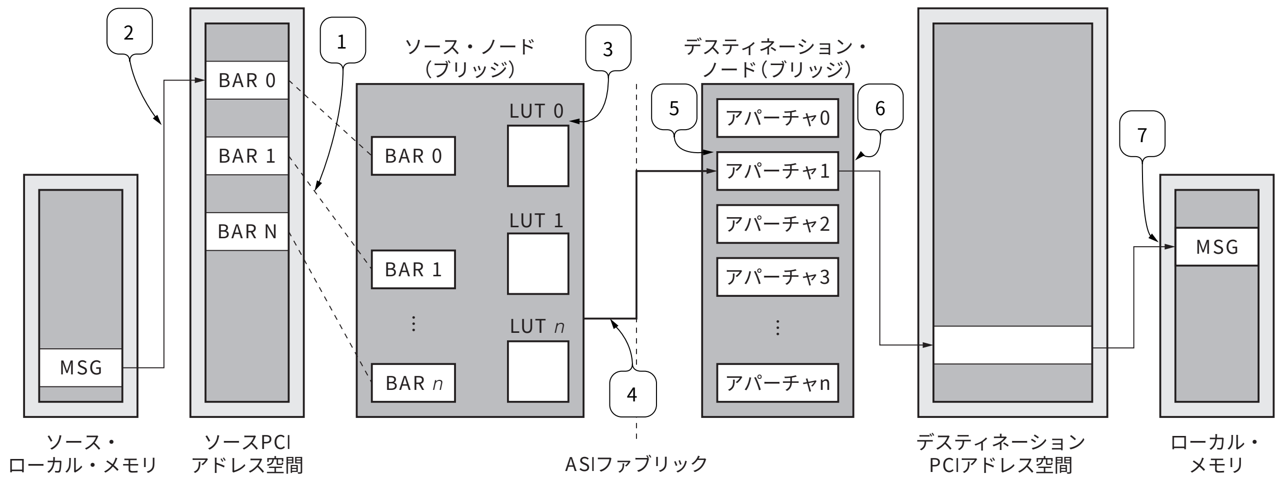


図15 SLSデータ・フローの一例

まず，ブリッジのBAR(base address register)をPCIアドレス空間に割り当てる(通常はOSが割り当てる)．次に，CPUがローカル・メモリにあるデータをブリッジに書き込み，ブリッジはPCIアドレスをデコードしてルックアップ・テーブル(LUT)からASIルート・ヘッダ情報を取得する．その後，ASIパケットを構成してファブリックに送出し，受信したパケットに対するプロテクション・チェックを行う．ブリッジはアパーチャ情報に基づきPCIライト(書き込み)トランザクションを発生し，このトランザクションによりデータがCPUのローカル・メモリへ書き込まれる．

常はファブリック・マネージメント・ソフトウェアを用意して管理します．

● データ転送プロトコルの規定

ASI-SIGは，前述のパケット・ルーティングのしくみに加えて，ファブリック管理用途を含めたいくつかの基本的なデータ転送プロトコルを規定しています(**表2**)．

1）ファブリック・マネージメント

ASI-SIGで規定しているPI-0，1，4，5は，おもにファブリックの管理に用いられます．これらのPIの中には，ASIデバイスの制御レジスタをアクセスするためのプロトコルや，ASIデバイス間のイベント通知プロトコルなどがあります．

2）PI-10：SLS(Simple Store Load)

SLSはリード/ライト方式によってデータを転送するプロトコルとして定義されているものです．

使用するASIデバイスにもよりますが，一般にCPUからSLSソース・デバイスにデータをPIO(programmed I/O)アクセスするか，SLSソース(データの送り出し側)デバイスのDMA(direct memory access)を用いてローカル・メモリのデータをパケット化し，SLSターゲット(データの受け取り側)デバイスに送信します．ターゲットとなるASIデバイスは少なくとも一つ(最大4,096)のアパーチャを持っています．ターゲットは，アパーチャが保持しているローカル・メモリ空間に対するベース・アドレス情報に，SLSパケット・ヘッダのアドレス・オフセット情報を加えたアドレスにペイロード内のデータを書き込むか，リードしたデータをペイロードに入れてソース・デバイスにパケットを返信します(**図15**)．

ASI-SIGは，ASIデバイスのローカル・バス側については言及していません．すなわち，ローカル・バス側は，例えばPCI Expressやそのほかのバスなど，特定のアーキテクチャに束縛されることはありません．

このほか，アドレス範囲がアクセスを許可されたものか，またアクセス・キーが一致しているかなど，権限を持たないソースからのアクセスから保護する機能もあります．

表2 プロトコル・インターフェース

PIインデックス	PI種別
0	パスの構築，マルチキャスト
1	輻そう管理
2	SAR(segmentation and reassembly)
3	ASI-SIGにて予約
4	デバイス管理
5	イベント通知
6～7	ASI-SIGにて予約
8～95	ASI-SIGにて規定 PI-8 PCI Express-ASIブリッジ PI-9 Socket Data Transfer PI-10 Simple Load/Store PI-11 Simple Queuing PI-12 Ethernet Protocol Interface(策定中)
96～126	メーカ定義PI
127	使用禁止

5

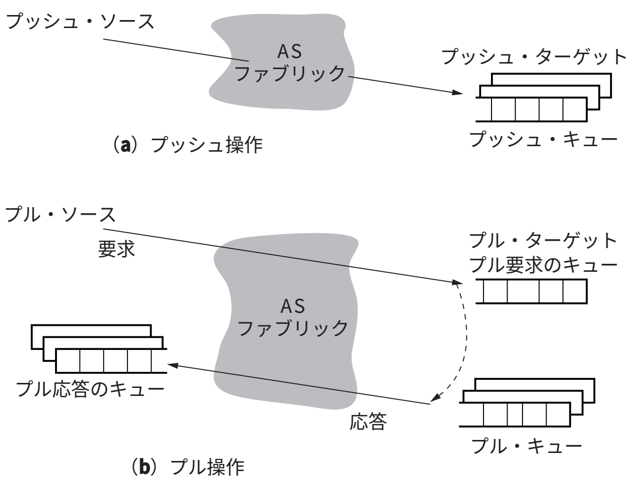


図16　SQの操作モデル
SQには，(a)のソースからターゲットへメッセージ・データを送信するプッシュ操作と，(b)のターゲットのキューからメッセージ・データを引き抜くプル操作がある．

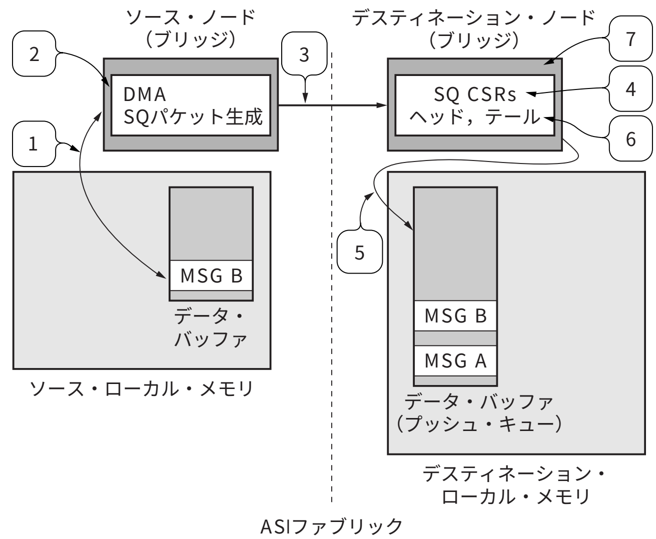


図17　SQデータフローの一例
ブリッジはDMAによりデータ・バッファからデータを読み出し，SQパケットを生成する．SQパケットをファブリックに送出すると，受信したSQパケットに対してプロテクション・チェックを行う．ブリッジは，受信したパケットを所定のデータ・バッファに書き込み，テール・ポインタを更新する．そして，デスティネーションのローカルCPUにデータが書き込まれたことを通知する．

3) PI-11：SQ(simple queuing)

SQは，メッセージングを目的としたプロトコルであり，プッシュ・オペレーションとプル・オペレーションの2通りの使用方法があります(**図16**)．プッシュ・キューはデータをエンドポイントに送信でき，プル・キューはエンドポイントのデータを順次アクセスできます．キューはインデックスによる識別が可能で，一つのASIエンドポイントで最大4,096個のキューを持てます(**図17**)．

そのほか，パケットの到着やキューの状態によりイベントを通知する機能やアクセス保護の機能などを備えています．また，SLSと同じくASIルート・ヘッダのPIが‘0’の場合，マルチキャストによるSQとなります．

4) PI-9：SDT(socket data transfer)

SDTは，ソケット・プログラミング・モデルをサポートするRDMA(remote direct memory access)プロトコルで，ASIエンドポイント間のデータ転送とそのためのハンドシェイクをプロセッサに負荷をかけずにハードウェアで行います．

SDTソースとSDTターゲットにはそれぞれ「ハンドル」があり，このハンドルがASIデバイスのSDT回路に対してASIエンドポイント間の接続情報を与えます．ハンドルには，ASIパケットを生成するために必要なルーティング情報やデータ・バッファに関する情報を保持するデスクリプタ情報が関連づけされています．各デバイスのSDT回路はこれらの情報に基づいてデータの転送を行います(**図18**)．

● ほかのプロトコルをパケットに入れて転送するしくみ

ASIでは，パケットの中にASI以外のプロトコル・データを包含して(これをプロトコル・カプセリングまたはカプセル化と呼ぶ)，ASIファブリック内を通過させ，送り先のエンドポイントで元のプロトコルに戻すことができます(このしくみをトンネリングと呼ぶ)．

これによって，ASIファブリック内にPCI ExpressやEthernet，SPI-4など，さまざまなプロトコルを混在させて，装置のデータ・プレーンやコントロール・プレーンを単一のASIファブリックに集約できます．

1) PI-8：PCI Expressブリッジ

PI-8を用いたプロトコル・カプセリングにより，PCI ExpressパケットにASIファブリック内を通過させます．このとき，パケット・ソース側のPCI Express-ASIブリッジは，PCI ExpressパケットにASIルート・ヘッダを付加して，PI-8パケットとしてASIファブリック内を通過させます．PI-8パケットを受け取ったASI-PCI Expressブリッジは，ASIパケットからASIルート・ヘッダを外してPCI Expressパケットを出力します(**図19**)．

このようにPI-8を用いることで両端のPCI Expressデバイスは，ASIファブリックの存在を意識することなく，パケットをやり取りできます．BIOSやOS，デバイス・ドライバなどは，従来使用しているPCI Express対応のものを

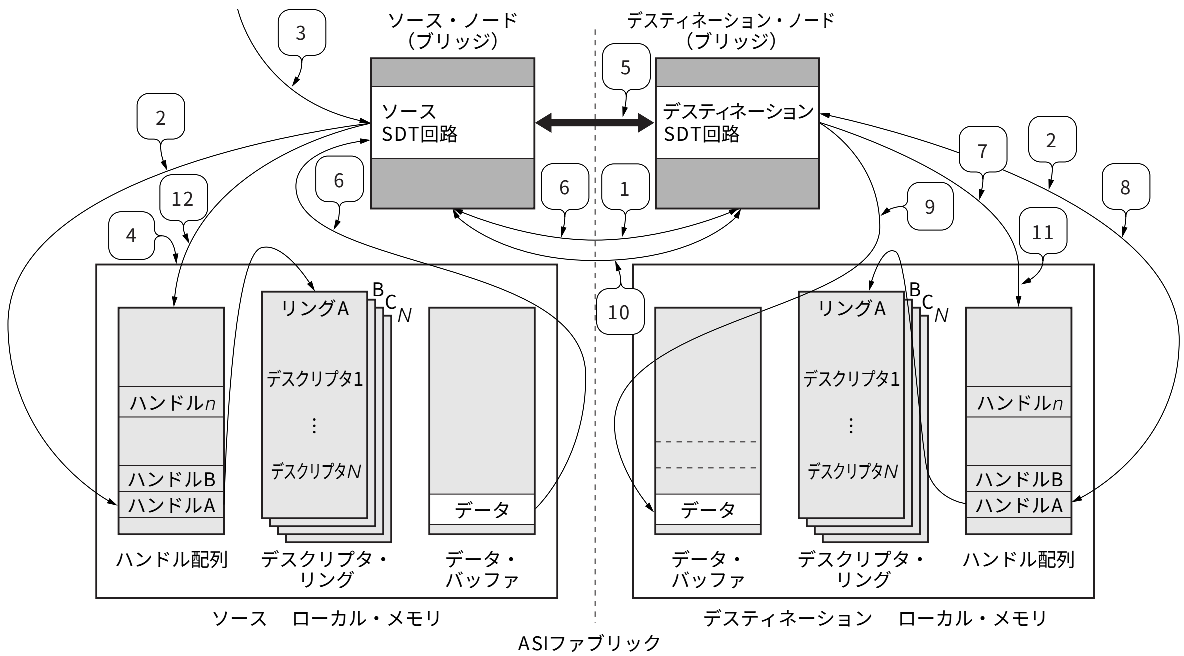


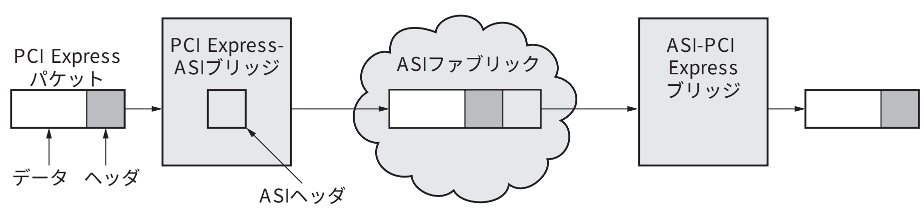
図18　SDTデータ・フローの一例

ソースとデスティネーションのCPUがそれぞれハンドルを設定する．また，これらのCPUはそれぞれのデスクリプタ・リングをローカル・メモリに設定する．その後，CPUがハンドルをSDT回路に指示する．SDT回路は指示されたハンドルから接続情報を取得する．ソースとデスティネーションのSDT回路は，ASIによる転送のネゴシエーションを行う．ソースSDT回路はデスクリプタとデータを読み出し，そのデータをSDTデータ・パケットによってASIファブリックに送出する．デスティネーションSDT回路は受信したSDTデータ・パケットから対応するハンドル情報とデスクリプタを読み出し，この情報に従ってデータをバッファに書き込む．その後，デスティネーションSDT回路はソース・ブリッジにアクノリッジ・パケットを送信し，ハンドル情報を更新する．ソースSDT回路はアクノリッジ・パケットを受け取ると，ハンドル情報を更新する．

5

図19
PCI Expressブリッジの概念

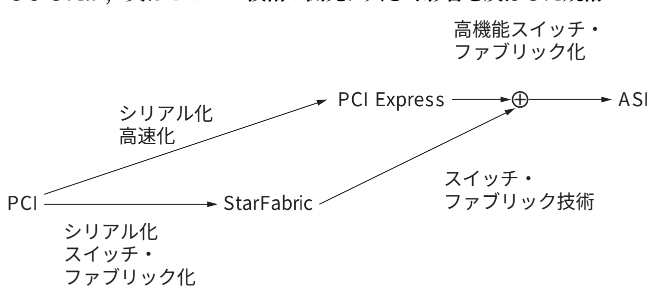
パケット・ソース側のPCI Express-ASIブリッジはPCI ExpressパケットにASIルート・ヘッダを付加し，PI-8パケットとしてASIファブリック内を通過させる．



Column　StarFabricとPCI Express，ASIの関係

ASIは，新しいインターコネクト標準規格としてASI-SIGが策定しましたが，実はこのASI技術の開発に大きく影響を及ぼした規格があります．それが，米国StarGen社が開発した，PCIバスをシリアル・スイッチ・ファブリック化するStarFabricです．対応デバイスが完成した2001年以降，日本をはじめ，欧米でも多くの採用実績があり，現在もPCIを使用する産業用機器でStarFabricが利用されています．

StarFabricは，PCIバスをシリアル・スイッチ・ファブリックにすることで拡張性や冗長性，QoSの向上，およびマルチプロセッシング環境への対応を実現します．単一リンクの速度はLVDSペア線当たり622MbpsとPCI Expressに及びませんが，通常は4ペアを使うことで2.5Gbpsの帯域を備えています．また，クレジット・ベースのフロー制御やトラフィック・クラス，仮想チャネルの考えかたなど，PCI Expressと共通の機能に加え，ソース・ベース・ルーティングやマルチキャスト，PI-8に相当するPCI-PCIブリッジ機能など，現在のASIの基本的な機能を実現しています(**図A-1**)．



図A-1　StarFabricとPCI Express，ASIの関係

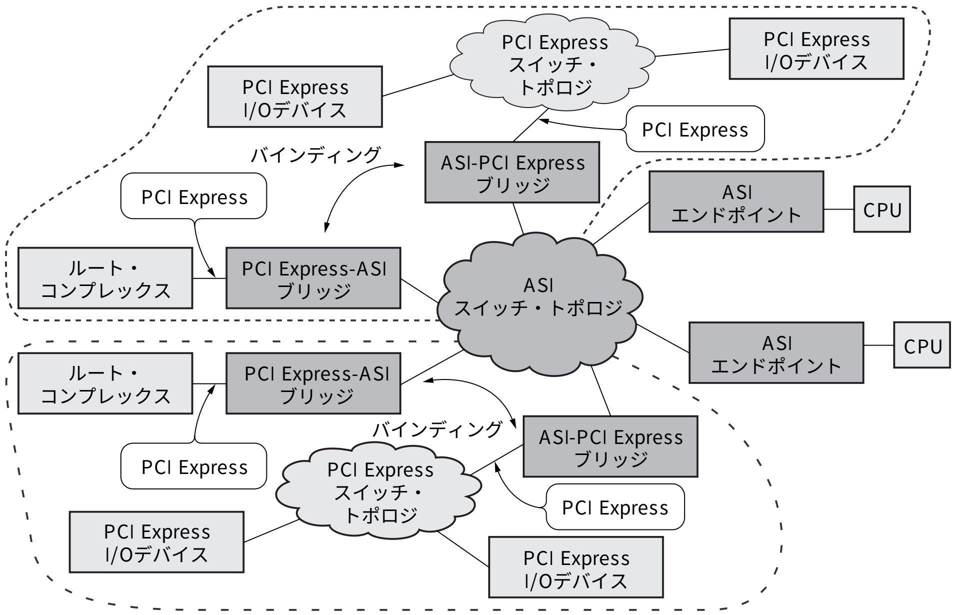


図20
PI-8バインディング
PI-8バインディングにより，ASIファブリック内に複数のPCI Expressドメインを構成できる．

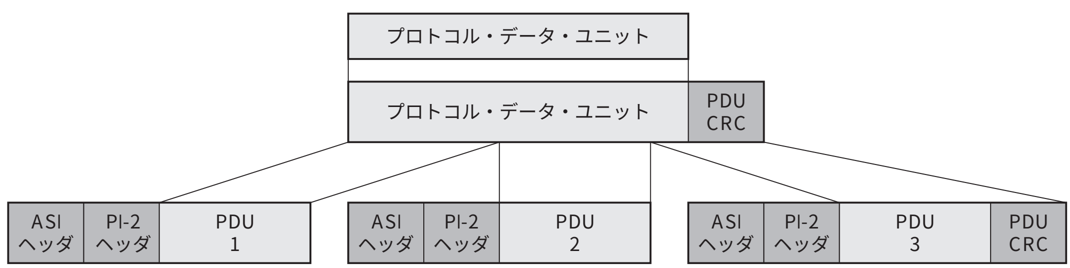


図21　SARによるセグメンテーション
ASIファブリックを通すプロトコルのパケット（PDU ： Protocol Data Unit）とそのCRCをASIパケット・サイズに分解し，分解情報を含むPI-2ヘッダを付加して，ASIパケットを構成する．PI-2パケットを受け取ったASIエンドポイントは，PI-2ヘッダ情報に基づき，分解されたパケットから元のPDUを再構成する．このSARにより，エンドポイントの先にあるデバイスは，ASIファブリックの存在を意識することなく，ASIファブリックを通信経路として利用できる．

そのまま使えます．

PI-8対応ブリッジでルート・コンプレックス側とエンドポイントの関連づけを行うことを「バインディング」と呼びます（**図20**）．ASIファブリックでは，複数組のPI-8バインディングを行えるため，単一のASIファブリック内を複数のPCI Express空間に分割できます．また，バインディングはソフトウェアを用いて動的に変更できます．例えば，CPUに障害が発生したとき，取り残されたI/Oをほかの CPUの配下に移動することも可能です．また，CPUの処理能力に応じてPI-8バインディングを変えれば，ハードウェアを変更する必要がなくなり，むだのない効果的な構成をとることもできます．

2）PI-2 ： SAR（segmentation and reassembly）

SARはトンネリングするプロトコルのパケットを分解（セグメント化）し，エンドポイントで組み立て直す（リアセンブリ）しくみです．これにより，トンネリングを行うプロトコルのパケット構造に影響を与えず，ASIファブリックを多様なプロトコルの通信経路として使用できます（**図21**）．

よこがわ・ともひで
（株）スターブリッジ

<筆者プロフィール>
横川与英．米国StarGen社のFAEとして国内のカスタマ・サポートや，ファブリック・ボード，ファブリック管理ソフトウェアの開発を手がける．2001年よりStarFabricの普及活動に携わる．当初は「わけのわからない技術」としてまともに相手にされなかったが，現在数多くのデザイン・ウィンを獲得し，今も増え続けている．2004年からはASIの普及活動を始めたが，いかに手短に，かつわかりやすく説明するかに苦悩している．